

High-order Virtual Element Method on polyhedral meshes

L. Beirão da Veiga^{a,b}, F. Dassi^a, A. Russo^{a,b}

^a*Department of Mathematics and Applications, University of Milano - Bicocca,
Via Cozzi 53, I-20153, Milano (Italy)*

^b*IMATI-CNR, 27100 Pavia (Italy)*

Abstract

We develop a numerical assessment of the Virtual Element Method for the discretization of a diffusion-reaction model problem, for higher “polynomial” order k and three space dimensions. Although the main focus of the present study is to illustrate some h -convergence tests for different orders k , we also hint on other interesting aspects such as structured polyhedral Voronoi meshing, robustness in the presence of irregular grids, sensibility to the stabilization parameter and convergence with respect to the order k .

Keywords: Virtual Element Method, polyhedral meshes, diffusion-reaction problem

2010 MSC: 65N30

1. Introduction

The Virtual Element Method (VEM) was introduced in [1, 2] as a generalization of the Finite Element Method (FEM) that allows for very general polygonal and polyhedral meshes, also including non convex and very distorted elements. The VEM is not based on the explicit construction and evaluation of the basis functions, as standard FEM, but on a wise choice and use of the degrees of freedom in order to compute the operators involved in the discretization of the problem. The adopted basis functions are virtual, in the sense that they follow a rigorous definition, include (but are not restricted to) standard polynomials but are *not* computed in practice; the accuracy of the method is guaranteed by the polynomial part of the virtual space. Using such approach introduces other potential advantages, such as exact satisfaction of linear constraints [3] and the possibility to build easily discrete spaces of high global regularity [4, 5]. Since its introduction, the VEM has shared a good degree of success and was applied to a large array of problems. We here mention, in addition to the ones above, a sample of papers [6, 7, 8, 9, 10, 11, 12, 13, 14] and refer to [15] for a more complete survey of the existing VEM literature.

Although the construction of the Virtual Element Method for three dimensional problems is accomplished in many papers, at the current level of de-

Email addresses: `lourengo.beirao@unimib.it` (L. Beirão da Veiga),
`franco.dassi@unimib.it` (F. Dassi), `alessandro.russo@unimib.it` (A. Russo)

velopment very few 3D numerical experiments are available in the literature [16, 13, 8]. Moreover, all these tests are limited to the lowest order case ($k = 1$).

The objective of this work is to numerically validate, for the first time, the VEM of general order k for three dimensional problems and show that this technology is practically viable also in this case. We consider a simple diffusion-reaction model problem in primal form and follow faithfully the construction in [1, 6] and the coding guidelines of [2]. Although the main focus of the present study is to illustrate some standard h -convergence tests for different orders k , it also hints on other aspects. In particular, we show some interesting possibilities related to polyhedral Voronoi meshing, we underline the robustness of the method in the presence of irregular grids, we investigate its sensibility to the stabilization parameter and consider also a convergence analysis in terms of the “polynomial” order k .

The paper is organized as follows. In Section 2 we introduce the model problem and the Virtual Element Method in three dimensions. The review of the method is complete but brief, and we refer to other contributions in the literature for a more detailed presentation of the scheme. Afterwards, in Section 3 an array of numerical tests are shown.

2. The Virtual Element discretization

In the present section we give a brief overview of the Virtual Element method in three space dimensions for the simple model problem of diffusion-reaction in primal form. More details on the method for this same model and formulation can be found in [1, 2, 6] while extension to variable coefficients is presented in [7]. In the following k will denote a positive integer number, associated to the “polynomial degree” of the virtual element scheme.

2.1. Notation

In the following, E will denote a polygon and P a polyhedron, while faces, edges and vertices will be indicated by f , e , and ν respectively.

If P is a polyhedron in \mathbb{R}^3 , we will denote by \mathbf{x}_P , h_P and $|P|$ the centroid, the diameter, and the volume of P , respectively. The set of polynomials of degree less than or equal to s in P will be indicated by $\mathcal{P}_s(P)$. If $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ is a multiindex, we will indicate by $m_{\boldsymbol{\alpha}}$ the scaled monomial

$$m_{\boldsymbol{\alpha}} = \left(\frac{x - x_P}{h_P} \right)^{\alpha_1} \left(\frac{y - y_P}{h_P} \right)^{\alpha_2} \left(\frac{z - z_P}{h_P} \right)^{\alpha_3} \quad (1)$$

and we will denote by $\mathcal{P}_s^{\text{hom}}(P)$ the the space of scaled monomials of degree exactly (and no less than) s :

$$\mathcal{P}_s^{\text{hom}}(P) = \text{span}\{m_{\boldsymbol{\alpha}} , |\boldsymbol{\alpha}| = s\} \quad (2)$$

where $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \alpha_3$. The case of a polygon $E \subset \mathbb{R}^2$ is completely analogous.

A face f of a polyhedron is treated as a two-dimensional set, using local coordinates (x, y) on the face. Edges of polyhedra and polygons are treated in an analogous way as one-dimensional set.

2.2. The model problem

Let $\Omega \subset \mathbb{R}^3$ represent the domain of interest (that we assume to be a polyhedron) and let Γ denote a subset of its boundary, that we assume for simplicity to be given by a union of some of its faces. We denote by $\Gamma' = \partial\Omega/\Gamma$.

We consider the simple reaction-diffusion problem

$$\begin{cases} -\Delta u + u = f & \text{in } \Omega \\ u = r & \text{on } \Gamma \\ \frac{\partial u}{\partial n} = g & \text{on } \Gamma' \end{cases} \quad (3)$$

where $f \in L^2(\Omega)$ and $g \in L^2(\Gamma')$ denote respectively the applied load and Neumann boundary data, and $r \in H^{1/2}(\Gamma)$ is the assigned boundary data function. The variational form of our model problem reads

$$\begin{cases} \text{Find } u \in H_\Gamma^1(\Omega) \text{ such that} \\ \int_\Omega \nabla u \cdot \nabla v + \int_\Omega uv = \int_\Omega fv + \int_{\Gamma'} gv \quad \forall v \in H_{\Gamma,0}^1(\Omega), \end{cases} \quad (4)$$

where

$$H_\Gamma^1 = \{v \in H^1(\Omega) : v|_\Gamma = r\}, \quad H_{\Gamma,0}^1 = \{v \in H^1(\Omega) : v|_\Gamma = 0\}.$$

2.3. Virtual elements on polygons

We start by defining the virtual element space on polygons. Given a generic polygon E , let the preliminary virtual space

$$\tilde{V}^k(E) = \left\{ v \in H^1(E) \cap C^0(E) : v|_e \in \mathcal{P}_k(e) \ \forall e \in \partial E, \ \Delta v \in \mathcal{P}_k(E) \right\}$$

with e denoting a generic edge of the polygon.

For any edge e , let the points $\{\nu_e^i\}_{i=1}^{k-1}$ be given by the $k-1$ internal points of the Gauss-Lobatto integration rule of order $k+1$ on the edge. We now introduce three sets of linear operators from $\tilde{V}^k(E)$ into real numbers. For all v in $\tilde{V}^k(E)$:

- evaluation of $v(\nu) \ \forall \nu$ vertex of E ; (5)

- evaluation of $v(\nu_e^i) \ \forall e \in \partial E, \ i = \{1, 2, \dots, k-1\}$; (6)

- moments $\int_E v p_{k-2} \ \forall p_{k-2} \in \mathcal{P}_{k-2}(E)$. (7)

The following projector operator $\Pi_E^\nabla : \tilde{V}^k(E) \rightarrow \mathcal{P}_k(E)$ will be useful in the definition of our space and also for computational purposes. For any $v \in \tilde{V}^k(E)$, the polynomial $\Pi_E^\nabla v \in \mathcal{P}_k(E)$ is defined by (see [1])

$$\begin{cases} \int_E \nabla(v - \Pi_E^\nabla v) \cdot \nabla p_k = 0 \quad \forall p_k \in \mathcal{P}_k(E) \\ \text{for } k = 1: \sum_{\substack{\nu \text{ vertex} \\ \text{of } E}} (v(\nu) - \Pi_E^\nabla v(\nu)) = 0, \\ \text{for } k \geq 2: \int_E (v - \Pi_E^\nabla v) = 0. \end{cases}$$

Note that, given any $v \in \tilde{V}^k(E)$ the polynomial $\Pi_E^\nabla v$ only depends on the values of the operators (5)-(6)-(7). Indeed, an integration by parts easily shows that the values of the above operators applied to v are sufficient to uniquely determine $\Pi_E^\nabla v$ and no other information on the function v is required [1, 2].

We are now ready to present the two dimensional virtual space:

$$V^k(E) = \left\{ v \in \tilde{V}^k(E) : \int_E v q = \int_E (\Pi_E^\nabla v) q \text{ for all } q \in \mathcal{P}_{k-1}^{\text{hom}}(E) \cup \mathcal{P}_k^{\text{hom}}(E) \right\}.$$

It is immediate to check that $\mathcal{P}_k(E) \subseteq V^k(E) \subseteq \tilde{V}^k(E)$. Moreover the following lemma holds (the proof can be found in [6, 7]).

Lemma 2.1. *The operators (5)-(6)-(7) constitute a set of degrees of freedom for the space $V^k(E)$.*

Finally note that, since any $p_k \in \mathcal{P}_k(E)$ can be written in an unique way as $p_k = p_{k-2} + q$, with $p_{k-2} \in \mathcal{P}_{k-2}(E)$ and $q \in \mathcal{P}_{k-1}^{\text{hom}}(E) \cup \mathcal{P}_k^{\text{hom}}(E)$, it holds

$$\int_E v p_k = \int_E v (p_{k-2} + q) = \int_E v p_{k-2} + \int_E (\Pi_E^\nabla v) q \quad (8)$$

for any $p_k \in \mathcal{P}_k(E)$. The first term in the right hand side above can be calculated recalling (7) while the second one can be computed directly by integration. This shows that we can actually compute $\int_E v p_k$ for any $p_k \in \mathcal{P}_k(E)$ by using only information on the degree of freedom values of v .

2.4. Virtual elements on polyhedrons

Let Ω_h be a partition of Ω into non-overlapping and conforming polyhedrons. We start by defining the virtual space V^k locally, on each polyhedron $P \in \Omega_h$. Note that each face $f \in \partial P$ is a two-dimensional polygon. Let the following boundary space

$$\mathcal{B}^k(\partial P) = \left\{ v \in C^0(\partial P) : v|_f \in V^k(f) \text{ for all } f \text{ face of } \partial P \right\}.$$

The above space is made of functions that on each face are two-dimensional virtual functions, that glue continuously across edges. Recalling Lemma 2.1, it follows that the following linear operators constitute a set of degrees of freedom for the space $\mathcal{B}^k(\partial P)$:

- evaluation of $v(\nu) \forall \nu$ vertex of P ; (9)

- evaluation of $v(\nu_e^i) \forall e$ edge of ∂P , $i = \{1, 2, \dots, k-1\}$; (10)

- moments $\int_f v p_{k-2} \forall p_{k-2} \in \mathcal{P}_{k-2}(f), \forall f$ face of ∂P . (11)

Once the boundary space is defined, the steps to follow in order to define the local virtual space on P become very similar to the two dimensional case. We first introduce a preliminary local virtual element space on P

$$\tilde{V}^k(P) = \left\{ v \in H^1(P) : v|_{\partial P} \in \mathcal{B}^k(\partial P), \Delta v \in \mathcal{P}_k(P) \right\}$$

and the “internal” linear operators

$$\bullet \quad \text{moments} \quad \int_P v p_{k-2} \quad \forall p_{k-2} \in \mathcal{P}_{k-2}(P). \quad (12)$$

We can now define the projection operator $\Pi_P^\nabla : \tilde{V}^k(P) \rightarrow \mathcal{P}_k(P)$ by

$$\left\{ \begin{array}{l} \int_P \nabla(v - \Pi_P^\nabla v) \cdot \nabla p_k = 0 \quad \forall p_k \in \mathcal{P}_k(P) \\ \text{for } k = 1: \quad \sum_{\substack{\nu \text{ vertex} \\ \text{of } P}} (v(\nu) - \Pi_P^\nabla v(\nu)) = 0, \\ \text{for } k \geq 2: \quad \int_P (v - \Pi_P^\nabla v) = 0. \end{array} \right. \quad (13)$$

An integration by parts and observation (8) show that the projection Π_P^∇ only depends on the operator values (9)-(10)-(11) and (12). Therefore we can define the local virtual space

$$V^k(P) = \left\{ v \in \tilde{V}^k(P) : \int_P v q = \int_P (\Pi_P^\nabla v) q \text{ for all } q \in \mathcal{P}_{k-1}^{\text{hom}}(P) \cup \mathcal{P}_k^{\text{hom}}(P) \right\}.$$

The proof of the following lemma mimicks the two dimensional case, see for instance [6].

Lemma 2.2. *The operators (9)-(10)-(11) and (12) constitute a set of degrees of freedom for the space $V^k(P)$.*

It is immediate to verify that $\mathcal{P}_k(P) \subseteq V^k(P)$, that is a fundamental condition for the approximation properties of the space. Moreover, again due to the observation above, the projection operator $\Pi_P^\nabla : V^k(P) \rightarrow \mathcal{P}_k(P)$ is computable only on the basis of the degree of freedom values (9)-(10)-(11) and (12). In addition, by following the same identical argument as in (8) we obtain that $\int_P v p_k$ is computable for any $p_k \in \mathcal{P}_k(P)$ by using the degrees of freedom. Therefore also the L^2 projection operator $\Pi_P^0 : V^k(P) \rightarrow \mathcal{P}_k(P)$, defined for any $v \in V^k(P)$ by

$$\int_P (v - \Pi_P^0 v) q_k = 0 \quad \forall q_k \in \mathcal{P}_k(P), \quad (14)$$

is computable by using the degree of freedom values.

Finally, the global virtual space $V^k \subset H^1(\Omega)$ is defined by using a standard assembly procedure as in finite elements. We define

$$V^k = \left\{ v \in H^1(\Omega) : v|_P \in V^k(P) \text{ for all } P \in \Omega_h \right\}.$$

The associated (global) degrees of freedom are the obvious counterpart of the

local ones introduced above, i.e.

- evaluation of $v(\nu) \forall \nu$ vertex of $\Omega_h \setminus \Gamma$; (15)

- evaluation of $v(\nu_e^i) \forall e$ edge of $\Omega_h \setminus \Gamma$, $i = \{1, 2, \dots, k-1\}$; (16)

- moments $\int_f v p_{k-2} \forall p_{k-2} \in \mathcal{P}_{k-2}(f), \forall f$ face of $\Omega_h \setminus \Gamma$; (17)

- moments $\int_P v p_{k-2} \forall p_{k-2} \in \mathcal{P}_{k-2}(P), \forall P \in \Omega_h$. (18)

2.5. Discretization of the problem

We start by introducing the discrete counterpart of the involved bilinear forms. Given any polyhedron $P \in \Omega_h$ we need to approximate the local forms

$$a_P(v, w) = \int_P \nabla v \cdot \nabla w, \quad m_P(v, w) = \int_P v w.$$

We follow [1, 2]. We first introduce the stabilization form

$$s_P(v, w) = \sum_{i=1}^{N_{\text{dof}}^P} \Xi_i(v) \Xi_i(w) \quad \forall v, w \in V^k(P), \quad (19)$$

where $\Xi_i(v)$ is the operator that evaluates the function v in the i^{th} local degree of freedom and N_{dof}^P denotes the number of such local degrees of freedom, see (9)-(10)-(11) and (12). We then set, for all $v, w \in V^k(P)$,

$$\begin{aligned} a_P^h(v, w) &= \int_P (\nabla \Pi_P^\nabla v) \cdot (\nabla \Pi_P^\nabla w) + h_P s_P(v - \Pi_P^\nabla v, w - \Pi_P^\nabla w), \\ m_P^h(v, w) &= \int_P (\Pi_P^0 v)(\Pi_P^0 w) + |P| s_P(v - \Pi_P^0 v, w - \Pi_P^0 w). \end{aligned} \quad (20)$$

The above bilinear forms are consistent and stable in the sense of [1]. The global forms are given by, for all $v, w \in V^k$,

$$a^h(v, w) = \sum_{P \in \Omega_h} a_P^h(v, w), \quad m^h(v, w) = \sum_{P \in \Omega_h} m_P^h(v, w).$$

Let now the discrete space with boundary conditions and its corresponding test space

$$V_\Gamma^k = \left\{ v \in V^k : v|_\Gamma = r_I \right\}, \quad V_0^k = \left\{ v \in V^k : v|_\Gamma = 0 \right\},$$

where r_I is, face by face, an interpolation of r in the virtual space $V^k(f)$.

We can finally state the discrete problem

$$\begin{cases} \text{Find } u_h \in V_\Gamma^k \text{ such that} \\ a^h(u_h, v_h) + m^h(u_h, v_h) = \int_\Omega f_h v_h + \int_{\Gamma'} g_h v_h \quad \forall v_h \in V_0^k, \end{cases} \quad (21)$$

where the approximate loading f_h is the L^2 -projection of f on piecewise polynomials of degree k , and where g_h is the L^2 -projection of g on piecewise polynomials (still of degree k) living on Γ' . Note that all the forms and operators

appearing above are computable in terms of the degree of freedom values of u_h and v_h .

We close this section by recalling a convergence result. The main argument for the proof can be found in [1], while the associated interpolation estimates where shown in [12] for two dimensions and extended in [17] to the three dimensional case.

Let now $\{\Omega_h\}_h$ be a family of meshes, satisfying the following assumption. It exists a positive constant γ such that all elements P of $\{\Omega_h\}_h$ and all faces of ∂P are star-shaped with respect to a ball of radius bigger or equal than γh_P ; moreover all edges $e \in \partial P$, for all $P \in \{\Omega_h\}_h$ have length bigger or equal than γh_P .

Theorem 2.3. *Let the above mesh assumptions hold. Then, if the data and solution is sufficiently regular for the right hand side to make sense, it holds*

$$\|u - u_h\|_{H^1(\Omega)} \leq Ch^{s-1} \left(|u|_{H^s(\Omega_h)} + |f|_{H^{s-2}(\Omega_h)} + |g|_{H^{s-3/2}(\Gamma'_h)} \right), \quad (22)$$

where $2 \leq s \leq k+1$, the real h denotes the maximum element diameter size and the constant C is independent of the mesh size. The norms appearing in the right hand side are broken Sobolev norms with respect to the mesh (or its faces).

The above result applies also if $1 \leq s < 2$, but in that case the regularities on the data f, g need to be modified. If the domain Ω is convex (or regular) then under the same assumptions and notations it also holds [7]

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^s \left(|u|_{H^s(\Omega_h)} + |f|_{H^{s-2}(\Omega_h)} + |g|_{H^{s-3/2}(\Gamma'_h)} \right). \quad (23)$$

Finally, we note that an extension of the results to more general mesh assumptions could be possibly derived following the arguments for the two-dimensional case shown in [18].

In the following numerical tests, we will in particular show the robustness of the method also for quite irregular meshes.

3. Numerical tests

In this section we collect the numerical results to evaluate the reliability and robustness of the Virtual Element Method in three dimensions.

3.1. Meshes and error estimators

Before dealing with the numerical examples, we define the domains where we solve the PDEs and the polyhedral meshes which discretize such domains. Moreover, we define the norms that we use to evaluate the error.

3.1.1. Meshes

We consider two different domains: the standard $[0, 1]^3$ cube, see Figure 1 (a), and a truncated octahedron [19], see Figure 1 (b).

We make different discretizations of such domains by exploiting the c++ library `voro++` [20]. More specifically we will consider the following three mesh types:

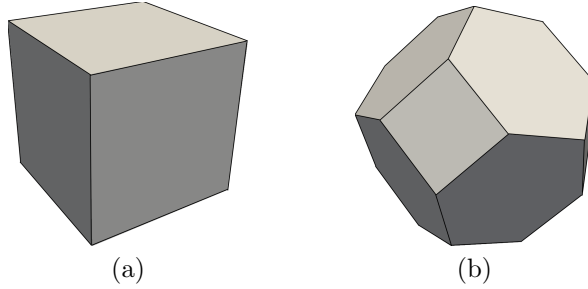


Figure 1: (a) Standard cube $[0, 1]^3$ and (b) the truncated octahedron.

Random refers to a mesh where the control points of the Voronoi tessellation are randomly displaced inside the domain. We underline that these kind of meshes are characterized by stretched polyhedrons so the robustness of the VEM will be severely tested.

CVT refers to a Centroidal Voronoi Tessellation, i.e., a Voronoi tessellation where the control points coincides with the centroid of the cells they define. We generate such meshes via a standard Lloyd algorithm [21]. In this case the Voronoi cells are more regular than the ones of the previous case.

Structured refers to meshes composed by structured cubes inside the domain and arbitrary shaped mesh close to the boundary. This mesh is built by considering as control points the vertices of a structured mesh of a cube \mathcal{C} (containing the input geometry \mathcal{G}) which are inside \mathcal{G} , see Figure 2 for a two dimensional example. When we consider the cube $[0, 1]^3$, this kind of mesh coincides with a structured mesh composed by cubes. These meshes are really interesting from the computational point of view. Indeed, the VEM local matrices are exactly the same for all cubes inside the domain. It is therefore possible to compute such local matrices *only* once so the computational effort in assembling the stiffness matrix as well as the right hand side is reduced. Moreover, the ensuring scheme may inherit some advantages of structural cubic meshes.

In Figure 3 we collect an example on the truncated octahedron geometry of all these kinds of meshes. To analyze the error convergence rate, we make a sequence of meshes with decreasing size for each mesh type.

3.1.2. Error norms

Let u be the exact solution of the PDE and u_h the discrete solution provided by the VEM. To evaluate how this discrete solution is close to the exact one, we use the local projectors of degree k on each polyhedron P of the mesh, $\Pi_P^\nabla u_h$ and $\Pi_P^0 u_h$, defined in (13) and (14), respectively. We compute the following quantities:

- **H^1 -seminorm error**

$$e_{H^1} := \sqrt{\sum_{P \in \Omega_h} |u - \Pi_P^\nabla u_h|_{H^1(P)}^2},$$

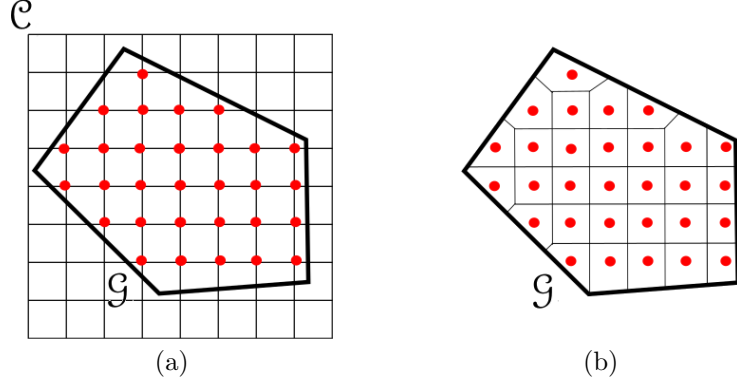


Figure 2: (a) The structured mesh of a square \mathcal{C} which contains the input geometry \mathcal{G} , where we highlight the vertices inside \mathcal{G} . (b) The final polyhedral Voronoi mesh where we highlight the control points.

- **L^2 -norm error**

$$e_{L^2} := \sqrt{\sum_{P \in \Omega_h} \|u - \Pi_P^0 u_h\|_{L^2(P)}^2},$$

- **L^∞ -norm error**

$$e_{L^\infty} := \max_{\nu \in \mathcal{N}} |u(\nu) - u_h(\nu)|,$$

where \mathcal{N} is the set of all the vertexes and internal edge nodes of the VEM scheme, see Equations (15) and (16). Since we do not take the max over all the domain but *only* on some nodes, e_{L^∞} is an approximation of the true L^∞ -norm. Moreover, in this case we can directly compute such quantity without resorting to the projections operators.

In the following subsections we will present some numerical tests to underline different computational aspects of the method. In all cases the mesh-size parameter h is measured in an averaged sense

$$h = \left(\frac{|\Omega|}{N_P} \right)^{1/3}, \quad (24)$$

with N_P denoting the number of polyhedrons in the mesh.

3.2. Test case 1: h -analysis for diffusion problem on a cube

Let us consider the problem

$$\begin{cases} -\Delta u &= f & \text{in } \Omega \\ u &= r & \text{on } \Gamma \\ \frac{\partial u}{\partial n} &= 0 & \text{on } \partial\Omega \setminus \Gamma \end{cases}, \quad (25)$$

where the domain Ω is the cube $[0, 1]^3$ and Γ is the union of the four faces corresponding to the planes $y = 0$, $y = 1$, $z = 0$ and $z = 1$. We choose the right

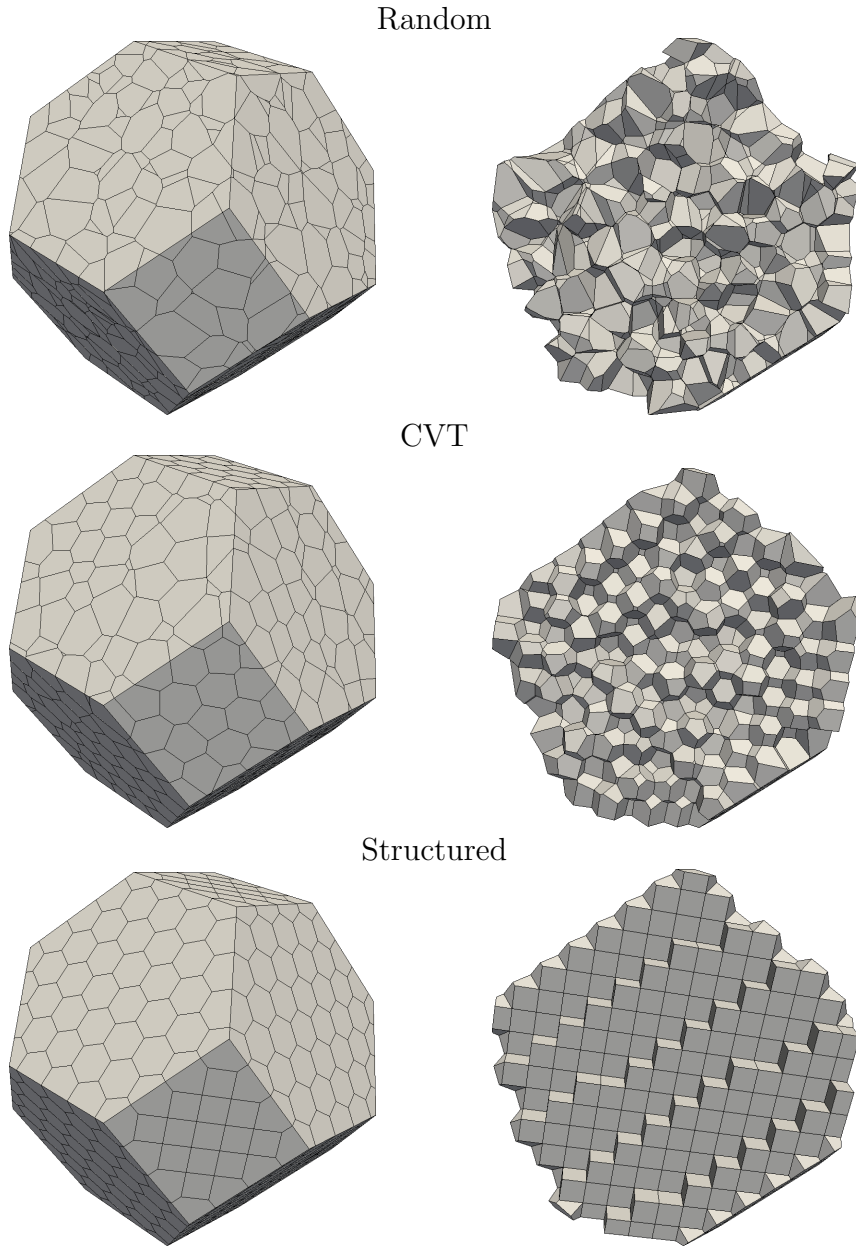


Figure 3: Different discretization of the truncated octahedron.

hand side f and r in such a way that the exact solution is

$$u(x, y, z) := \sin(\pi x) \cos(\pi y) \cos(\pi z).$$

In this example we consider all the three types of discretizations introduced in Subsection 3.1, i.e., Random, CVT and Structured. Note that in this case the last type of discretization becomes a standard structured cubic mesh.

In Figure 4 we show the resulting graphs and in Tables 1 and 2 we provide the convergence rates. These data show that we achieve the theoretical convergence rate for all the VEM approximation degrees and for each type of meshes, see equations (22), (23).

	convergence rates		
mesh type	$k = 1$	$k = 2$	$k = 3$
Structured	1.0344	2.0543	3.0125
Random	1.0927	2.0465	3.0656
CVT	1.0492	2.0672	3.0970

Table 1: Test case 1: H^1 -seminorm convergence rates.

	convergence rates		
mesh type	$k = 1$	$k = 2$	$k = 3$
Structured	1.9763	3.2551	4.0372
Random	1.9136	3.1067	4.0678
CVT	2.0230	3.2144	4.4620

Table 2: Test case 1: L^2 -norm convergence rates.

3.3. Test case 2: h -analysis for diffusion-reaction problem on a polyhedron

In this example we consider the problem

$$\begin{cases} -\Delta u + u = f & \text{in } \Omega \\ u = r & \text{on } \Gamma = \partial\Omega \end{cases}, \quad (26)$$

where the domain Ω is the truncated octahedron, see Figure 1 (b), and we choose the right hand side f and r in such a way that the exact solution is

$$u(x, y, z) := \sin(2xy) \cos(z).$$

In this example we analyze the convergence rate for VEM approximation degrees from 1 to 3, and compare the results obtained with the three mesh types described in Subsection 3.1.1.

In Figure 5 we plot the convergence graphs with respect to the total number of degrees of freedom N_{dof} . Considering that (for fixed order k and for fixed mesh family) the mesh-size parameter is expected to behave as $h \sim N_{\text{dof}}^{-1/3}$, it follows that both error norms behave as expected from the theory (see (22) and (23)).

Moreover we observe that the error is slightly affected by the shape of the mesh elements. Indeed, the errors associated with the Random mesh are always larger than the ones obtained with a more regular mesh, while the Structured meshes yield the best results (even when compared to the CVT meshes).

3.4. Test case 3: convergence analysis with different k

In this example we consider the convergence with respect to the accuracy degree k . We fix the truncated octahedron geometry and we solve the following

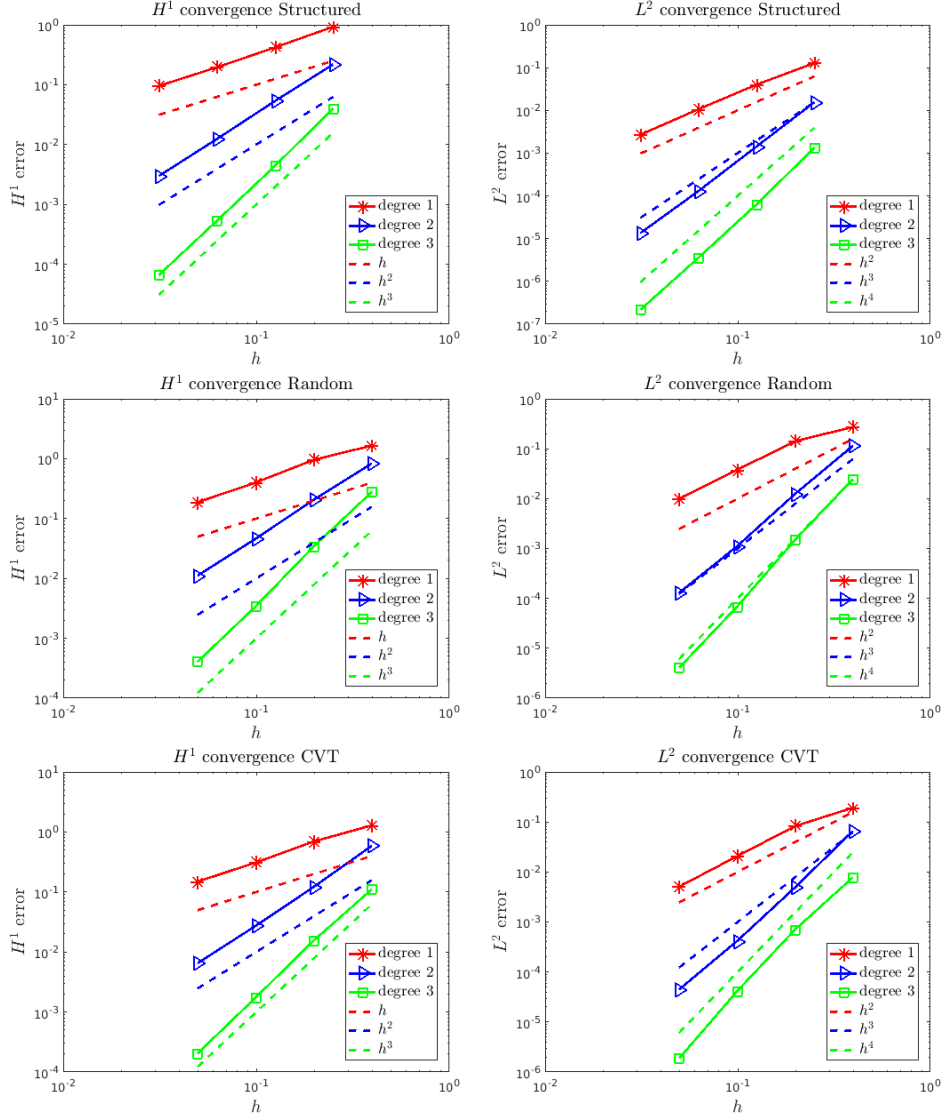


Figure 4: Test case 1: h -convergence with different meshes.

problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = r & \text{on } \Gamma = \partial\Omega \end{cases}, \quad (27)$$

where the right hand side f and the boundary condition r are chosen in such a way that the exact solution is

$$u(x, y, z) := \sin(\pi x) \cos(\pi y) \cos(\pi z).$$

The mesh is kept fixed (the CVT mesh of the truncated octahedron composed by 116 polyhedrons) and we rise the polynomial degree k from 1 to 5. In Figure 6

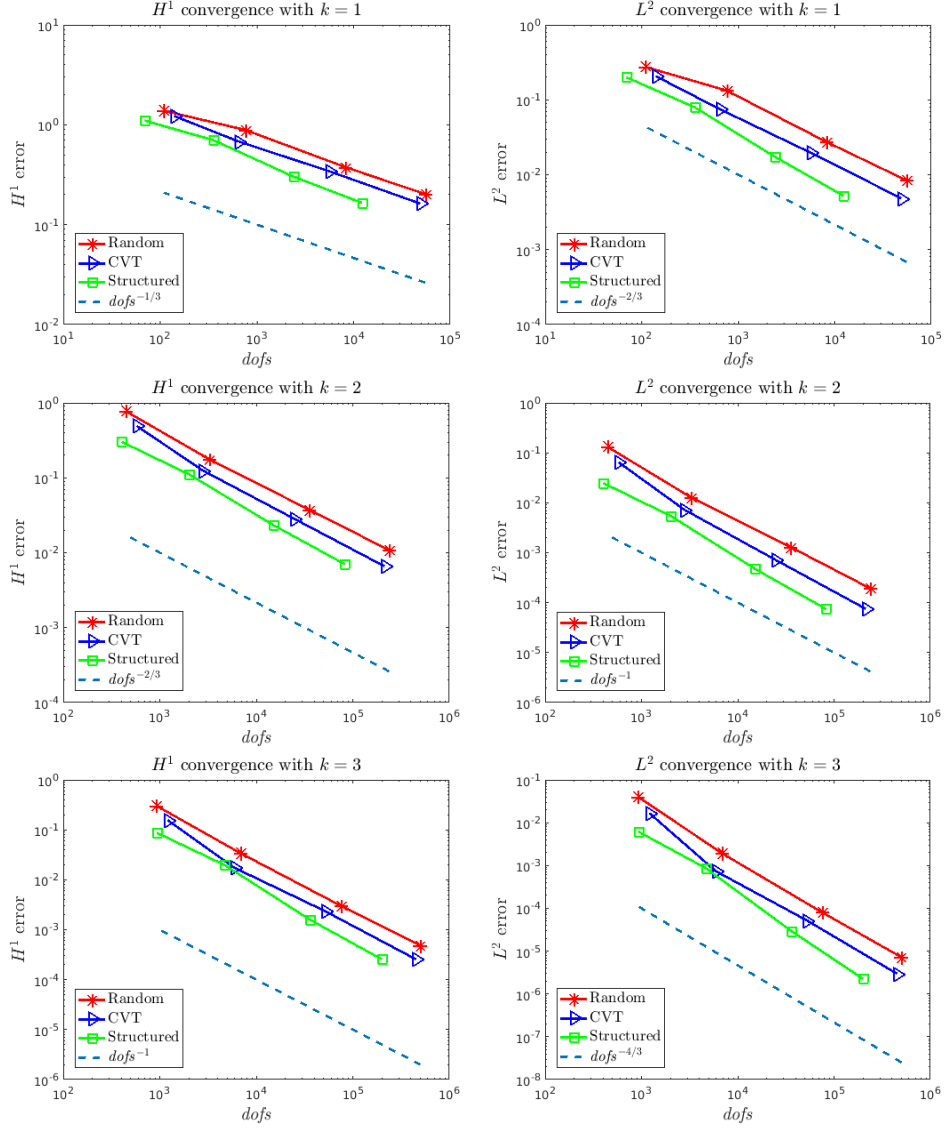


Figure 5: Test case 2: $dofs$ -convergence with different meshes.

we provide the convergence graphs of both H^1 -seminorm and L^∞ -norm. The trend of these errors show an exponential convergence in terms of k and are thus aligned with the existing theory for the two-dimensional case [22]. However, both the H^1 and the L^∞ errors show a slight bend in the convergence graphs for $k=5$. This behavior is probably due to the stabilizing matrix (19) that should be better devised in order to develop a spectral approximation strategy.

Although such aspect deserves a deeper study, that is beyond the scopes of the current paper, we propose a novel stabilization strategy which, at least in the present context, cures the problem.

Let $\{\varphi_i\}_{i=1}^{N_P^P}$ represents the canonical basis functions on element P , defined

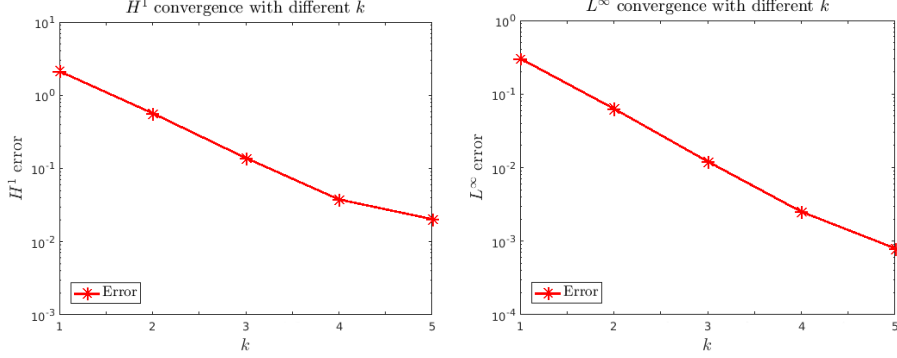


Figure 6: Test case 3: convergence with respect to the order k .

by

$$\varphi_i \in V^k(P), \quad \Xi_j(\varphi_i) = \delta_{ij} \text{ for } j = 1, 2, \dots, N_{\text{dof}}^P,$$

where we refer to (19) for the notation. Then, a diagonal stabilization form $s_P(\cdot, \cdot)$ should satisfy (see (20) and [1, 2])

$$h_P s_P(\varphi_i, \varphi_i) \simeq a_P(\varphi_i, \varphi_i) \quad i = 1, 2, \dots, N_{\text{dof}}^P$$

in order to mimic the original energy $a_P(\cdot, \cdot)$ of the basis functions. The original form in (19) corresponds to assuming $a_P(\varphi_i, \varphi_i) \simeq h_P$, that (considering the involved scalings) can be shown to be a reasonable choice with respect to h_P (and thus for moderate k). On the other hand, choice (19) may be less suitable for a higher k , especially in three dimensions, where different basis functions may carry very different energies. We therefore propose a simple alternative (still using a diagonal stabilizing form)

$$\tilde{s}_P(\varphi_i, \varphi_i) = \max\{h_P, a_P(\Pi_P^\nabla \varphi_i, \Pi_P^\nabla \varphi_i)\} \quad i = 1, 2, \dots, N_{\text{dof}}^P.$$

Note that at the practical level computing $a_P(\Pi_P^\nabla \varphi_i, \Pi_P^\nabla \varphi_i)$ is immediate, since such term is simply the i^{th} term on the diagonal of the consistency matrix $a_P(\Pi_P^\nabla \varphi_i, \Pi_P^\nabla \varphi_j)$, that is already computed.

The choice above, referred in the following as *diagonal recipe*, cures the problem, see Figure 7. This fact is further confirmed by the data in Table 3. Here, we compute the slopes of these lines at each step and we numerically show that the recipe diagonal stabilization yields better results.

3.5. Test case 4: patch test

In [1, 2] it is shown that VEM passes the so-called “patch test”. If we are dealing with a PDE whose solution is a polynomial of degree k and we use a VEM approximation degree equal to k , we recover the “exact solution”, i.e., the solution up to the machine precision.

We make a patch test for VEM approximation degrees k from 1 up to 5. More specifically, we consider the following PDE

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = r & \text{on } \Gamma = \partial\Omega \end{cases} \quad (28)$$

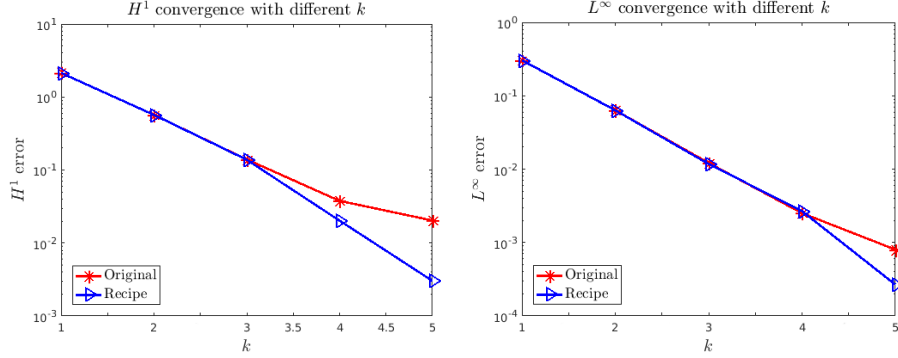


Figure 7: Test case 3: convergence with respect to the order k , comparison between the original and the diagonal recipe stabilizations.

	H^1 convergence			
2-5 Original	-1.3219	-1.4065	-1.3045	-0.6195
Recipe	-1.3218	-1.4144	-1.9211	-1.8977

	L^∞ convergence			
Original	-1.5704	-1.6532	-1.5736	-1.1455
Recipe	-1.5698	-1.6850	-1.4805	-2.3065

Table 3: Test case 3: slopes at each step for the original and the recipe diagonal stabilization.

where the right hand side f and the Dirichlet boundary condition r are chosen in accordance with the exact solution

$$u(x, y, z) := (x + y + z)^k.$$

Since we are not interested in varying the mesh size but *only* the VEM approximation degree, we run the experiments on the same coarse CVT mesh of the truncated octahedron composed by 116 polyhedrons.

In Table 4 we collect the results. The errors are close to the machine precision, but for higher VEM approximation degrees they become larger. This fact is natural and stems from the conditioning of the matrices involved in the computation of the VEM solution. Indeed, as in standard FEM, their condition numbers become larger when we consider higher VEM approximation degrees.

3.6. Test case 5: stabilizing parameter analysis

In this example we make an analysis on the stabilizing part of the local stiffness matrix. We slightly modify a_P^h defined in Subsection 2.5 by introducing the parameter $\tau \in \mathbb{R}$, i.e.,

$$a_P^h(v, w) = \int_P (\nabla \Pi_P^\nabla v) \cdot (\nabla \Pi_P^\nabla w) + \tau h_P s_P(v - \Pi_P^\nabla v, w - \Pi_P^\nabla w). \quad (29)$$

solution degree	H^1 -seminorm	L^2 -seminorm	L^∞ -norm
1	5.9775e-12	6.1919e-13	1.1479e-12
2	2.2008e-11	1.8416e-12	4.9409e-12
3	1.0490e-10	1.1284e-11	8.5904e-12
4	3.0959e-10	1.0039e-10	2.6197e-11
5	1.1563e-09	6.9693e-09	1.5433e-10

Table 4: Test case 4: patch tests errors with the mesh CVT of the truncated octahedron composed by 116 elements.

We fix a standard Poisson problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \Gamma = \partial\Omega \end{cases}, \quad (30)$$

where $\Omega = [0, 1]^3$ and whose exact solution is

$$u(x, y, z) := \sin(\pi x) \cos(\pi y) \cos(\pi z).$$

We exploit the same discretization of Ω , a CVT mesh composed by 1024 polyhedrons, and solve this problem for many values of the parameter τ . More specifically, we consider $\tau = 10^t$ for one hundred uniformly distributed values of t in $[-2, 2]$ and we compute the errors for the corresponding VEM solutions in the H^1 -seminorm and L^∞ -norm.

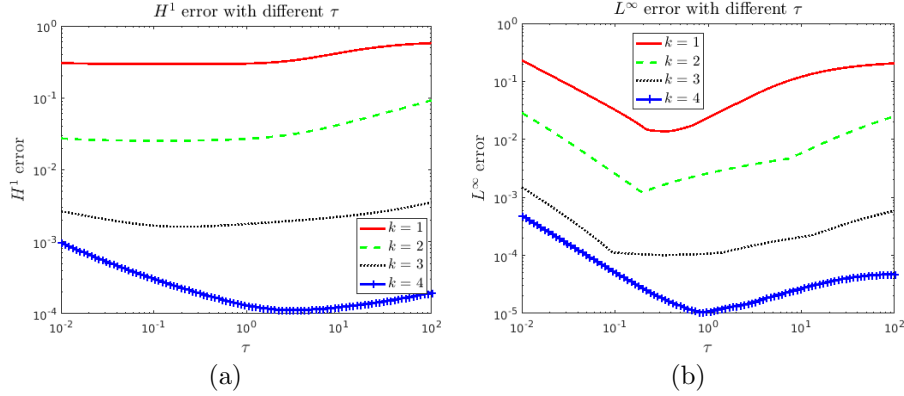


Figure 8: Test case 5: (a) H^1 error and (b) L^∞ error for varying τ and different VEM approximation degrees k .

In Figure 8 we provide the graphs of both errors, as a function of τ in a logarithmic scale, for VEM approximation degree $k = 1, 2, 3$ and 4. We can observe that the trend of the errors is the same for each k : it grows whenever very small or very large choices are taken for the τ parameter. Although the L^∞ -norm seems more sensible than the H^1 -seminorm, the method appears in general quite robust with respect to the parameter choice. For instance, we

show in Table 5 the quantities

$$\delta_{e_{H^1}} := \frac{\max_{\tau \in [10^{-1}, 10]} (e_{H^1})}{\min_{\tau \in [10^{-1}, 10]} (e_{H^1})}, \quad \delta_{e_{L^\infty}} := \frac{\max_{\tau \in [10^{-1}, 10]} (e_{L^\infty})}{\min_{\tau \in [10^{-1}, 10]} (e_{L^\infty})},$$

representing the ratios of maximum to minimum error for the parameter range $[10^{-1}, 10]$, that corresponds to a factor of 100 between minimum and maximum τ . From this table we can appreciate that the error ratios are all within an acceptable range.

k	$\delta_{e_{H^1}}$	$\delta_{e_{L^\infty}}$
1	1.4487e+00	8.9624e+00
2	1.7014e+00	4.9538e+00
3	1.3884e+00	2.0908e+00
4	2.6989e+00	4.8393e+00

Table 5: Test case 5: Ratios of the errors for different degrees k . Parameter $\tau \in [10^{-1}, 10]$.

Acknowledgments

The first and second authors have received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement no. 681162).

References

- [1] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, A. Russo, Basic principles of Virtual Element Methods, *Math. Models Methods Appl. Sci.* 23 (2013) 119–214.
- [2] L. Beirão da Veiga, F. Brezzi, L. D. Marini, A. Russo, The hitchhikers guide to the virtual element method, *Math. Models Methods Appl. Sci.* 24 (8) (2014) 1541–1573.
- [3] L. Beirão da Veiga, C. Lovadina, G. Vacca, Divergence free virtual elements for the stokes problem on polygonal meshes, *ESAIM Math. Model. Numer. Anal.* 51 (2) (2017) 509–535.
- [4] F. Brezzi, L. Marini, Virtual Element Method for plate bending problems, *Comput. Methods Appl. Mech. Engrg.* 253 (2012) 455–462.
- [5] P. Antonietti, L. Beirão da Veiga, S. Scacchi, M. Verani, A C^1 virtual element method for the Cahn-Hilliard equation with polygonal meshes, *SIAM Journal on Numerical Analysis* 54 (1) (2016) 34–56.
- [6] B. Ahmed, A. Alsaedi, F. Brezzi, L. Marini, A. Russo, Equivalent Projectors for Virtual Element Methods, *Comput. Math. Appl.* 66 (3) (2013) 376–391.
- [7] L. Beirão da Veiga, F. Brezzi, L. D. Marini, A. Russo, Virtual element methods for general second order elliptic problems on polygonal meshes, *Math. Models Methods Appl. Sci.* 26 (4) (2016) 729–750.

- [8] H. Chi, L. Beirao da Veiga, G. Paulino, Some basic formulation of the virtual element method (VEM) for finite deformations, *Comput. Meth. Appl. Mech. Engrg.* 318 (09) (2017) 148–192.
- [9] B. Ayuso, K. Lipnikov, G. Manzini, The nonconforming virtual element method, *ESAIM: M2AN* 50 (3) (2016) 879–904.
- [10] M. F. Benedetto, S. Berrone, A. Borio, S. Pieraccini, S. Scialò, A hybrid mortar virtual element method for discrete fracture network simulations, *Journal of Computational Physics* 306 (2016) 148 – 166.
- [11] E. Caceres, G. Gatica, A mixed virtual element method for the pseudostressvelocity formulation of the stokes problem, *IMA J. of Numer. Anal.* 37 (1) (2017) 296–331.
- [12] D. Mora, G. Rivera, R. Rodríguez, A virtual element method for the Steklov eigenvalue problem, *Math. Models Methods Appl. Sci.* 25 (8) (2015) 1421–1445.
- [13] A. Gain, G. Paulino, S. Leonardo, I. Menezes, Topology optimization using polytopes, *Comput. Methods Appl. Mech. Engrg.* 293 (2015) 411–430.
- [14] G. Vacca, Virtual element methods for hyperbolic problems on polygonal meshes, *Comput. Math. Appl.* doi:<http://dx.doi.org/10.1016/j.camwa.2016.04.029>.
- [15] L. Beirão da Veiga, A. Chernov, L. Mascotto, A. Russo, Exponential convergence of the hp virtual element method with corner singularities (2016). [arXiv:1611.10165](https://arxiv.org/abs/1611.10165).
- [16] A. Gain, C. Talischi, G. Paulino, On the Virtual Element Method for Three-Dimensional Elasticity Problems on Arbitrary Polyhedral Meshes, *Comp. Meth. in Appl. Mech. and Engrng.* 282 (2014) 132 – 160.
- [17] A. Cangiani, E. Georgoulis, T. Pryer, O. Sutton, A posteriori error estimates for the virtual element method, Preprint [arXiv:1603.05855](https://arxiv.org/abs/1603.05855).
- [18] L. Beirao da Veiga, C. Lovadina, A. Russo, Stability analysis for the virtual element method, Preprint [arXiv:1607.05988](https://arxiv.org/abs/1607.05988).
- [19] R. Williams, *The Geometrical Foundation of Natural Structure: A Source Book of Design*, Dover Publications, 1979.
- [20] C. H. Rycroft, Voropp: A three-dimensional voronoi cell library in c++, *Chaos* 19 (4) (2009) 041111.
- [21] Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: Applications and algorithms, *SIAM Rev.* 41 (4) (1999) 637–676.
- [22] L. Mascotto, L. Beirao da Veiga, A. Chernov, A. Russo, Basic principles of hp virtual elements on quasiuniform meshes, *Math. Mod. and Meth. in Appl. Sci.* 26 (08) (2016) 1567–1598.